

Energy Efficient Ant Colony Algorithms for Data Aggregation in Wireless Sensor Networks

Chi Lin^a, Guowei Wu^a, Feng Xia^a, Mingchu Li^a, Lin Yao^a, Zhongyi Pei^a

^a*School of Software, Dalian University of Technology, China*

Abstract

In energy-constrained wireless sensor networks, energy efficiency is critical for prolonging the network lifetime. A family of ant colony algorithms called DAACA for data aggregation are proposed in this paper. DAACA consists of three phases: initialization, packets transmissions and operations on pheromones. In the transmission phase, each node estimates the remaining energy and the amount of pheromones of neighbor nodes to compute the probabilities for dynamically selecting the next hop. After certain rounds of transmissions, the pheromones adjustments are performed, which take the advantages of both global and local merits for evaporating or depositing pheromones. Four different pheromones adjustment strategies which constitute DAACA family are designed to prolong the network lifetime. Experimental results indicate that, compared with other data aggregation algorithms, DAACA shows higher superiority on average degree of nodes, energy efficiency, prolonging the network lifetime, computation complexity and success ratio of one hop transmission. At last, the features of DAACA are analyzed.

Keywords: wireless sensor networks, data aggregation, ant colony optimization, energy efficiency, network lifetime

1. Introduction

The main task of wireless sensor network (WSN) [1, 2] is to detect and report the events of the physical world. In most cases, nodes are batteries powered [3] with limited energy resources. Suppose that when a node runs out of power and stops working, the original transmission paths will be changed. Nodes nearby will suffer from heavier work, because of sharing responsibility of the exhausted node which casts heavy burden of them. The energy dissipating rate of these nodes will become faster. This process spreads which will cause the packets loss or even network congestion. Moreover, the performance of the network depends on the persistence of the sensors to a large extent. Hence, the main challenge for the energy-constrained network is to design energy-efficient routing

Email addresses: wgdut@dlut.edu.cn (Guowei Wu), f.xia@acm.org (Feng Xia)

protocols [4, 5, 6, 7, 8] which guarantees the persistence and balances the energy consumption of the network.

In recent years, methods of data aggregation [9] are attracting the attentions of the researchers. A large number of works of data aggregation protocols in wireless sensor networks for energy preservation have been published.

To begin with, we will provide a brief background of the data aggregation methods of WSN. The key idea is to combine the data coming from different sources which eliminates redundancy [10], minimizes the number of transmissions and thus saves energy. This method shifts the focus from the traditional address centric approaches for networking to a data centric approach.

Some existing methods [11, 12, 13, 14] focus on establishing and maintaining topology of the data aggregation tree in which data packets are transmitted and aggregated to specific nodes. In [11], the authors propose a method which generates Minimum Spanning Tree (MST) by implementing Prim algorithm from a global view. In [12], each node locally forms MST to establish the aggregation topology to eliminate energy costs. In [13], the authors integrate LMST with RNG[15] to establish the network topology. In [14], the authors put forward an ant colony based data aggregation method which takes advantages of pheromones to heuristically set up the network topology. In the aforementioned methods, packets are transmitted in pre-allocated paths and aggregated in some specific nodes. Nevertheless, such topologies consume much energy in maintenance. Nodes in the aggregation tree are actually facing large energy consumption. The common problems of the existing methods are:

1. The energy consumptions of constructing and maintaining network topologies are high.
2. Issues of balancing energy consumption[16, 17] are not considered.

The ant colony optimization(ACO)[18] is one of the most useful swarm intelligence[19] which has been successfully applied in many optimization problems such as TSP[20], CVRP[21] as well as routing in wireless sensor networks[22, 23, 24]. In this paper, a measure of data aggregation based on ant colony optimization is proposed which overcomes shortcomings mentioned in [9, 25, 26, 27, 28]. We expand the ant colony algorithms [29] by proposing a family of energy-efficient Data Aggregation Ant Colony Algorithms (DAACA for short), which contains Basic-DAACA(i.e. the Basic algorithm of DAACA), ES-DAACA(i.e. Elitist Strategy based DAACA), MM-DAACA(i.e. Max-Min based DAACA) and ACS-DAACA(i.e. Ant Colony System based DAACA).

DAACA include three phases: 1) the initialization, 2) packets transmissions and 3) operations on pheromones. Initially, all the sensors are deployed randomly, then they set up their routing tables in a self-organization manner. We consider each node as an artificial ant which is raised in ant colony optimization for solving the global optimization problem. The packets transmitted between nodes are considered as tools for updating or adjusting the pheromones. Each node estimates the remaining energy and the amount of pheromones to compute the probabilities for dynamically selecting the next hop. After certain rounds

of transmissions, the adjustments of pheromones are performed, which originally combines the advantages of global and local adjustments for evaporating or depositing pheromones. Four different pheromones adjustment strategies are designed to achieve the global optimal in prolonging network lifetime.

The contributions of the paper can be summarized as follows:

1. We proposed a family of ant colony algorithms for data aggregation which aims at saving energy and prolonging network lifetime.
2. We devise the evaporating and the depositing pheromones approaches which take advantages of global and local merits.
3. We design three heuristic methods(i.e. ES-DAACA,MM-DAACA,ACS-DAACA) for improving the performance of the basic algorithm(i.e. Basic-DAACA) which specialize the way of evaporating and depositing pheromones.
4. We develop a platform to compare the characteristics(e.g. average energy cost, network lifetime, average degree and so on) of DAACA family with other data aggregation methods(e.g. PEDAP, PEDAP-PA, L-PEDAP and so on).

The remainder of the paper proceeds as follows. Section 2 gives an overview of the state of the art of data aggregation in wireless sensor networks. Section 3 introduces the preliminaries of the paper. Section 4 presents the system model and defines the problem. Section 5 describes DAACA in details. The simulation results and performance analysis are presented in section 6. Section 7 concludes the paper.

2. Related Work

In literatures, many algorithms are proposed to prolong the lifetime of the network based on data aggregation [30, 31, 16, 11, 12, 13, 14]. Some references are concentrating on establishing energy efficient topologies to save energy.

In [11], an algorithm named PEDAP (Power Efficient Data gathering and Aggregation Protocol) is proposed, which generates Minimum Spanning Tree (MST) by utilizing Prim algorithm. It is a near optimal minimum spanning tree based routing scheme. But it cannot guarantee the energy efficiency of the network, then an energy-aware version of PEDAP, namely PEDAP-PA is proposed to maximize the network lifetime. However, the sink node needs to periodically broadcast and calculate the MST of the network, thereby, the workload of the sink node is high.

In [12], a Local Minimum Spanning Tree algorithm called LMST is presented to establish the network topology, although it can effectively reduce the average degree of nodes, some prominent problems still emerge. Each node needs to periodically calculate and update the MST(Minimum Spanning Tree) locally which leads to a high computational overhead for each node. Moreover, each node needs to communicate with its neighbors to obtain the energy condition of neighbors, which still costs much energy.

In [13], a localized, self organizing, robust and energy efficient data aggregation algorithm named L-PEDAP is proposed which combines LMST with RNG

[15]. Although it is proved to have the capability of prolonging the network lifetime, its topology construction procedure is nearly identical with that of LMST, hence, it cannot be considered as an energy-efficient algorithm.

To reduce the cost of constructing and maintaining the energy-efficient topology, a heuristic method called ACA (Ant Colony Algorithm) which is based on ant colony algorithm is proposed in [14]. In the ant colony optimization, a colony of artificial ants is used to construct solutions guided by the pheromones trails and heuristic information [32]. This behavior enables ants to find the shortest paths between the food source and the nest in a random manner. The author initiatively designs the rules for depositing and evaporating pheromones. However, the requirements of depositing pheromones can be easily met. When depositing pheromones, each node needs to communicate with its neighbors. Therefore there are lots of communications within the pheromones adjusting period, which leads to much energy cost overhead. Therefore, it cannot be considered as an energy-efficient algorithm either.

Moreover, although some other methods do not depend on energy efficient topologies to save energy, some literatures are still valuable to be mentioned. In [31], the authors describe the design and implementation of a running system for energy-efficient surveillance which allows a group of cooperating sensor to detect and track the positions of moving vehicles. That method can trade off the energy-awareness and surveillance performance by adjusting the sensitivity of the system. In [30], the data aggregation is motivated by optimizations of aggregation queries. The authors present variety of techniques to improve the reliability and performance of the proposed solutions. In [16], the authors explore two methods to further reduce energy consumption in the context of network aggregation in sensor networks. Firstly, a group-aware network configuration method is designed, which groups the sensors into clusters. Secondly, a framework to use temporal coherency tolerances in conjunction with aggregation to save energy is proposed.

3. Preliminaries

In this section, we illustrate related knowledge in this paper.

3.1. Ant Colony Optimization

The idea of Ant Colony Optimization is originated by Marco Dorigo[33], in observing the exploitation of seeking shortest path between their nest to the food source although limited cognitive abilities are owned by ants. In Ant Colony Optimization, artificial ants are implemented to search for heuristic solutions of the problems. Originally, artificial ants are utilized to find the solutions stochastically based on the probabilities whose calculations are defined according to certain circumstances(i.e. according to the characteristic of the problem). After a feasible solution is constructed, the evaporation and releasing of pheromones are executed which dynamically update the amount of pheromones. Therefore,

the solution will gradually approach to the global optimal solution of the problem. This procedure repeats until certain number of ants have finished their trails. The process of ACO can be summarized as Algorithm 1.

Algorithm 1 ACO Pseudocode

- 1: Initialization of data structure and parameters
 - 2: **while** terminate conditions **do**
 - 3: Construct solutions
 - 4: Update statistics
 - 5: Update pheromones
 - 6: **end while**
-

In general, we can summarize the advantages of ant colony optimization.

1. ACO is a probabilistic mechanism for solving computational problems which can be reduced to finding good paths in graphs.
2. The solution of ACO will gradually approach to the global optimal solution in terms of adjusting the quantities of pheromones.
3. In ACO, each ant is considered to have individually limited cognitive ability.
4. The process of ACO is the cooperation of ants which gradually find optimal solutions in graph.

In wireless sensor networks, a large number of nodes are deployed. To find the global optimal prolong lifetime solution can be transformed into forming data aggregation tree by global optimization. Hence, firstly, our aim is to find a solution which can approach to global optimization. Secondly, to prolong the lifetime of the network, we need to balance the energy cost of each node, therefore, we need to establish dynamical topology of the networks. Thirdly, each node has limited computation capability, it only hold the local information(i.e. the information of neighbors). What's more, the network can be abstracted as a graph where the vertexes stand for the nodes and the edges represent the link between nodes. Last but not least, researches of the ACO in wireless sensor networks[14, 34, 35] lay a good foundation for our work. From the analysis above, we conclude that ACO is suitable for data aggregation in wireless sensor network. Therefore, we employ ant colony optimization in resolving data aggregation in wireless sensor network.

3.2. Ant Colony System

Ant Colony System(ACS) applies the mechanisms of ACO but some changes has been made to overcome the drawbacks of ACO and enhance the performance of ACO. The main differences can be summarized as:

1. It applies a more active behavior, which can make full use of the accumulated pheromones in the path.
2. The evaporating and releasing of the pheromones are only carried out in the most optimal path so far.

3. When an ant passes through a trail, the pheromone of that trail will be reduced which aims at enhancing the possibility of finding more optimal solutions in other trails.

Inspired by the features of ACS, we implement the ant colony system into our data aggregation method in section 5.5 to enhance the performance.

3.3. ACA Algorithm

In [14], the authors propose an data aggregation(ACA). First, each compute the hop count to the sink node. When the source node wants to send data, it will select the next hop node by the probability which mainly depends on hop count of the neighbor nodes to the sink.

$$p_k(i, j) = \frac{\tau(i, j) \times \eta(i, j)^\beta}{\sum_{u \in N_i} \tau(i, u) \times \eta(i, u)^\beta}$$

where $\tau(i, j)$ is the pheromones level from node i to node j , $\eta(i, j)$ is the inverse of the hop count from node j to the sink node adding one, N_i is the number of neighbors of the node i , and β is a parameter which determines the relative influence of heuristic values $\eta(i, j)$, $\beta > 0$. Then the authors design the message communication mechanism which increase the probability of intersection of the routing path to promote the results of data aggregation. Finally, the evaporation and releasing of pheromones are defined which dynamically adjust the quantities of pheromones in the paths. A node who has a lower hop count and happen to exist in the path are more likely to get pheromone which enhances the probability of obtaining messages in later transmissions. Whereas a node cannot receive a message for more and certain round, the pheromones of such node will be evaporate, which reduce the possibility of receiving of nodes of such nodes.

To aggregate size of packets, in this paper, we develop a family of ant colony based data aggregation algorithm DAACA. The similarities and differences between DAACA family and ACO can be summarized as follows: **Similarities:**

1. Ant Colony Optimization methods are employed which aims at establishing data aggregation tree.
2. Methods of releasing and evaporating of pheromones are developed.
3. The ultimate targets are to save energy.

Differences:

1. DAACA focus on balancing the energy cost of the whole network to prolong the network lifetime whereas ACO does not take the lifetime into account.
2. When selecting the next hop, the remaining energy, the distance between the neighbor to the sender are considered which aims at fairly selecting the next hop based on energy cost in DAACA, whereas ACA only cares about the distance and the quantities of the pheromones.

Table 1: Energy Dissipated Table

| Symbol | Meaning | Energy Dissipated |
|---------------------|-------------------------|-------------------|
| $E_{Tx-elec}$ | Transmitter Electronics | 50nJ/bit |
| $E_{Rx-elec}$ | Receiver Electronics | 50nJ/bit |
| ε_{amp} | Transmit Amplifier | 100J/bit/ m^2 |

3. In ES-DAACA, MM-DAACA and ACS-DAACA, we take advantages of accumulated pheromones to increase the probability of selecting the global optimal solution. But ACO does not take any actions to obtain global optimal solutions.
4. In MM-DAACA and ACS-DAACA, the range of the quantities of pheromones are constrained which aims at avoiding trapping into local optimal solutions whereas ACO does not limit the ranges of pheromones.
5. In ACS-DAACA, we reduce the probabilities of the nodes who locates in the aggregation tree to diversify the solutions to get more optimal solutions(i.e. the data aggregation tree). On the contrary, ACO can not guarantee the diversity of the solutions.

3.4. Energy Consumption Model

Many energy models [27] are used for energy consumption in wireless sensor networks. In our work, we employ the model in which a packet with the size k to be transmitted in the distance of d will consume:

$$E_{Tx}(k, d) = E_{Tx-elec} \times k + \varepsilon_{amp} \times k \times d^2 \quad (1)$$

for the transmitter and

$$E_{Rx}(k) = E_{Rx-elec} \times k \quad (2)$$

for the receiver respectively. The definitions of the symbols are listed in table 1.

4. System Model

We consider a scenario where the sensor nodes are homogenous and energy-constrained. All the nodes (including the sink node) are stationary and randomly distributed. Further, replenishing energy via replacing batteries on hundreds of nodes (in possibly harsh terrains) is infeasible. The basic operation in such a system is the systematic gathering of sensed data to be eventually transmitted to a base station for processing. The key challenge in such data gathering is conserving the sensor energies, so as to maximize their lifetimes.

4.1. Definitions and Assumptions

In this section, some definitions and assumptions are defined below.

4.1.1. Definition

Remaining Energy: The initial energy minus the energy consumed in transmission process.

The remaining energy reflects the usage of energy. Moreover it can indicate when the network exhausts.

Energy Difference: The maximum remaining energy minus the minimum remaining energy in the network.

The energy difference indicates the balanced usage of energy in the network. Small energy difference shows the balanced usage of energy of all nodes in the network and vice versa.

4.1.2. Assumption

Assumption 1: In one aggregation procedure, n packets with the size of k will be eventually compressed into one with the size of k .

If the packet size is bigger than k , it will be sliced into pieces with constant size of k . If the size is smaller than k , it will be enlarged into size of k . This assumption is identical with that of [32]

Assumption 2: Only the transmission process costs energy.

Since we mainly focus on the energy cost in data aggregation to prolong the lifetime, therefore we omit effects of the energy costs of nodes of their own (e.g. calculating, waiting and so on). We assume that the energy cost can only happen when transmitting or receiving packets.

4.2. Problem Statement

The network is modeled as a visibility graph $G = (V, E)$, where V is the set of sensor nodes. Each node has the maximum transmission range denoted by R , by which it can setup its neighbor set and routing table. e_{ij} represents the distance between node i and node j which is smaller than R , meanwhile the neighbor set (denoted as $Nb(i)$) of node i indicates a set of nodes whose distances are less than R (i.e. $Nb(i) = \{j | e_{ij} \leq R\}$). The set of e_{ij} comprises the E . Each source periodically senses and collects the data from the surroundings, then sends the data to the next hop until the data reaches the sink node. The goal is to find an energy efficient routing path to prolong the network lifetime. The best way to solve this problem is to aggregate the multiple data packets and reduce the traffic in the network. Thus, multiple transmission paths should be merged into smaller ones to relieve the energy consumptions of data transmissions. Multiple sources periodically send data to their neighbors until the data packets reach the sink node in a hop by hop manner.

5. Ant Colony Algorithms for Data Aggregation

Overall we propose the system architecture of DAACA. Then we design a basic algorithm, after that three heuristic algorithms are proposed to enhance the performance of the basic algorithm.

Since each algorithm in DAACA family has the same algorithm structure, differences only exist in the process of evaporating and depositing pheromones. We firstly illustrate algorithm structure of DAACA family and then introduce each algorithm in details. The symbols used in DAACA are illustrated in table 2.

Table 2: Related Definitions

| Symbol | Notations |
|------------------------|---|
| R | The maximum transmission range. |
| id | The identity of the node. |
| $Nb(i)$ | The neighbor nodes of node i . |
| $round$ | The transmission times. |
| $roundToUpdate$ | The number of rounds to update the pheromones. |
| $p(i, j)$ | The probability for node i to select node j as the next hop. |
| $\eta(i, j)$ | The pheromone of node i to node j . |
| $E'_{distance}(i, j)$ | The energy distance from node i to node j . |
| $E_{distance}(i, j)$ | The Euclidian distance between node i and node j . |
| $e_1(i)$ | The current value of node i divided by E_{init} . |
| $e_2(i, j)$ | The estimation energy of node i to node j divided by E_{init} . |
| e_{ij} | The distance between node i and node j . |
| $E_{cur}(i)$ | The current energy of node i . |
| $E_{estimate}(i, j)$ | The evaluation energy of node i to node j . |
| E_{init} | The initial energy of each node. |
| $\tau(i, j)$ | The inverse value of $E'_{distance}(i, j)$ |
| $Times(i, j)$ | The transmission times from node i to node j . |
| M_{PCost} | The minimum energy cost in a path. |
| η_{Max} | Upper bound of pheromones. |
| η_{Min} | Lower bound of pheromones. |
| $conj(i)$ | The conjunction times of node i . |
| $source$ | The source node(s for short) of the network. |
| $destination$ | The destination node (d for short) of the network. |
| $current$ | The current node(c for short) who receives the data packet. |
| Continued on next page | |

Table 2 – continued from previous page

| Symbol | Notations |
|-----------|---|
| $next$ | The next node(n for short)which current will send the packet to. |
| P_{min} | The minimum energy cost path. |

5.1. DAACA Algorithm Structure

Similarly, in wireless sensor network, nodes are considered as the artificial ants. The amount of the pheromones of links are recorded in the routing table, whenever a node receives a packet, it will figure out the next hop based on the information of the pheromones, the distance and the remaining energy according to the routing table. Then each node will update the information. After certain rounds of transmissions, procedure of evaporating and depositing pheromones will be applied which help DAACA to adjust the amount of the pheromones. The objective of this process is to guide the aggregation routing path to approach to the global optimal routing path which will conserve energy for the network to some extent. Next we will illustrate the realizations of DAACA in details.

Initially every node constructs its own neighbor set and routing table by broadcasting its *id* (i.e. the identity of the node) and location information within R . Afterwards, the transmission process starts. Packets are sent from the source nodes to the sink node in each round. When a node receives a packet, it will evaluate contents of the routing table (e.g. the remaining energies, the amount of pheromones) to calculate the transmission probabilities of selecting the next hop. Then one of its neighbors will be selected. When the number of *round* equals to the *roundToUpdate*, all nodes will perform the evaporating and depositing operations. Each node evaporates the pheromones of its neighbor and deposits pheromones according to different specific conditions (referring to section 5.2- 5.5 in the following sections). After that each node will update the pheromones of its neighbors.

Algorithm 2 illustrates the structure of DAACA. Firstly, the initialization of the network is carried out. Each node broadcasts "hello" message to its neighbors. Then only the nodes who are nearer to the sink node will be recorded in the routing table and the contents of the table are initialized. Then the transmission begins, the source nodes periodically send the data packets to the sink node. When a relay node receives a packet, it will calculate the probability of selecting the next hop based on the elements in the routing table (e.g the distance, the amount of the pheromones, estimated remaining energy et.al). The transmission proceeds until the data are forwarded to the sink node (i.e. line 3-line 18). After transmitting *roundToUpdate* rounds, adjustments of pheromones are carried out (i.e. line 20-line 22). Each node updates the routing table to keep the information fresh. At last each node broadcasts its actual energy, by which the neighbor nodes will update of the contents of the routing table (i.e. line 23-line 24).

Algorithm 2 DAACA Structure

```
1: Network Initialization
   Node Initialization
   Neighbor Initialization
   Routing Table Initialization
2: The source node begin to send its data packets to the destination hop-by-
   hop.
3: for all  $t \in Nbr(s)$  do
4:    $s$  evaluates the energy of  $t$ .
5:    $s$  calculates  $p(s, t)$ .
6:    $s$  selects the next hop node  $n$  based on  $p(s, t)$ 
7: end for
8:  $s \rightarrow n$  //  $s$  sends the packets to  $n$ 
9: while  $s \neq d$  do
10:   $c = n$ 
11:   $c$  aggregates data packets
12:  for all  $t \in Nbr(c)$  do
13:     $c$  evaluates the energy of  $t$ .
14:     $c$  calculates  $p(c, t)$ .
15:     $c$  selects the next hop node  $n$  based on  $p(c, t)$ 
16:  end for
17:   $c \rightarrow n$ 
18: end while
19:  $round = round + 1$ 
20: if  $round = roundToUpdate$  then
21:   Evaporating Pheromones.
22:   Depositing Pheromones.
23:   Updating Routing Table.
24:   Energy Broadcasting.
25: end if
```

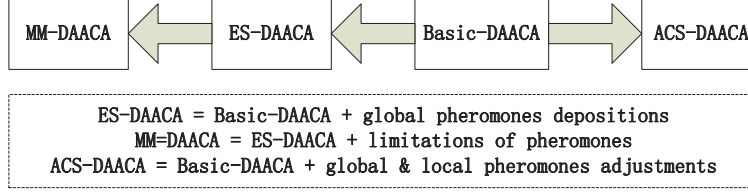


Figure 1: The relationship between Basic-DAACA, ES-DAACA, MM-DAACA, ACS-DAACA

In the following sections, we present four ant colony heuristic algorithms (i.e. Basic-DAACA, ES-DAACA, MM-DAACA, ACS-DAACA) for adjusting the pheromones especially the evaporating and depositing of pheromones. Basic-DAACA is the basic algorithm, other algorithms are designed based on Basic-DAACA. The relationship between Basic-DAACA, ES-DAACA, MM-DAACA and ACS-DAACA can be formalized in figure 1.

ES-DAACA adds additional global pheromones depositions into Basic-DAACA. MM-DAACA limits the ranges of pheromones in ES-DAACA. ACS-DAACA adds both global and location pheromones adjustments in Basic-DAACA. Followings are explicit illustrations of DAACA family.

5.2. Basic-DAACA

Basic-DAACA is the basic algorithm of DAACA family. For the pheromone is the most critical part in adjusting the probabilities in the routing table, we firstly illustrate the constitution of the routing table. The routing table contains the following elements: $\{ id, E'_{distance}(i, j), E_{estimate}(i, j), \eta(i, j), p(i, j) \}$, $j \in Nb(i)$. id refers to the identity of the node. $p(i, j)$ is the probability for node i to select node j as the next hop which is computed as follows:

$$p(i, j) = \frac{\tau(i, j)^\alpha \times \eta(i, j)^\beta}{\sum_{j \in Nbr(i)} [\tau(i, j)^\alpha \times \eta(i, j)^\beta]} \quad (3)$$

α and β are two parameters which determine the relative influence of $\tau(i, j)$ and $\eta(i, j)$. $\tau(i, j)$ represents the inverse value of the energy distance $E'_{distance}(i, j)$ between node i and j , which holds:

$$\tau(i, j) = \frac{1}{E'_{distance}(i, j)} \quad (4)$$

$E'_{distance}(i, j)$ can be computed as:

$$E'_{distance}(i, j) = \frac{E_{distance}(i, j)}{e_1(i) \times e_2(i, j)} (0 < e_1 < 1, 0 < e_2 < 1) \quad (5)$$

Where:

$$e_1(i) = \frac{E_{cur}(i)}{E_{init}} \quad (6)$$

$$e_2(i, j) = \frac{E_{estimate}(i, j)}{E_{init}} \quad (7)$$

$E_{distance}(i, j)$ is the energy distance which can be calculated as:

$$E_{distance}(i, j) = E_{Tx-elec} \times k + \varepsilon_{amp} \times k \times e_{ij}^2 \quad (8)$$

k is the size of the packet. $E_{estimate}(i, j)$ denotes the energy of node j estimated by node i , which can be estimated according to:

$$E_{estimate}(i, j) = E_{init} - \frac{E_{init} - E_{estimate}(i, j)}{Times(i, j)} \times [Times(i, j) + 1] \quad (9)$$

When a node receives a packet, it will evaluate the remaining energy of all the neighbors and updates all the values in the routing table to dynamically select the next hop.

When *round* is multiple of *roundToUpdate*, procedure of adjusting the pheromones starts. First, the pheromones should be evaporated. We utilize the equation (10) to evaporate pheromones.

$$\eta(i, j) = (1 - \rho) \times \eta(i, j) \quad (10)$$

ρ stands for the fraction of pheromones that are not evaporated. Then the procedure of depositing pheromones is performed. Each node selects the neighbor with the maximum estimation energy (e.g. the node j), and increases the pheromone of node j by $E_{distance}(i, j)$ as:

$$\eta(i, j) = \eta(i, j) + E_{distance}(i, j) \quad (11)$$

When *round* is multiple of *roundToUpdate*, algorithm 3 is called to adjust the pheromones for each node. Firstly, the evaporating and depositing of pheromones are taken (i.e. line 1 - line 9). Then updating of the routing table are carried out (i.e. line 11 - line 14).

Once algorithm 3 is finished, in the next period of *roundToUpdate*, the node with the highest $E_{distance}(i, j)$ will have a higher probability of being selected as the next hop. After finishing depositing pheromones, the process of updating the estimated energy value is performed. Each node will broadcast its current energy in the range of R . The value of $E_{estimate}(i, j)$ will be updated.

If a node happens to exist in the conjunction of two or more different paths, the parent nodes will also use equation (11) to increase the pheromones.

Basic-DAACA algorithm avoids the computing overhead of repeatedly constructing or maintaining the network topology, which saves energy and prolongs the network lifetime. In the following sections, some heuristic methods are proposed to enhance the performance of Basic-DAACA.

Algorithm 3 Adjusting pheromones of Basic-DAACA

```
1: for all  $i \in V$  do
2:    $i$  evaporates the pheromones according to equation (10).
3:    $i$  searches for the node with the highest  $E_{estimate}(i, j)$  where  $j \in Nb(i)$  in
     neighbor set.
4:    $i$  deposits pheromones according to equation (11).
5:   if  $conj(i) \geq 2$  then
6:     for all  $t \in Nb(i)$  do
7:        $t$  deposits pheromones of  $i$  according to equation (11).
8:        $i$  broadcasts the current energy.
9:     end for
10:  end if
11:  if  $i$  receives a broadcast message from  $j$  then
12:     $i$  updates  $E_{estimate}(i, j)$ .
13:     $i$  updates the routing table.
14:  end if
15: end for
```

5.3. Elitist Strategy based DAACA(ES-DAACA)

In order to establish better aggregation paths, the global pheromones deposition is employed. We call this heuristic method Elitist Strategy based DAACA(ES-DAACA for short). We apply another format of the packet header, which obtains the *id* sequence of nodes and total energy consumption as: *ID_List*, *Energy_Consumption*. *ID_List* records all the nodes in the trace from the source to the current node. *Energy_Consumption* is defined as the total transmission energy costed from the source to the current node.

Once a packet is received by a node, algorithm 4 will be taken which adds *id* information and energy consumption information into the newly defined packet header. Owing to the header, the sink node can find the minimum cost aggregation path and adjust the amount of pheromones to obtain more optimal solutions.

Algorithm 4 Packet receiving process of ES-DAACA

```
1: if  $i$  receives the data packet then
2:    $i$  selects the next hop  $j$ .
3:    $i$  adds its id information into ID_List.
4:    $i$  adds  $E_{distance}(i, j)$  into the energy consumption field.
5: end if
```

When a packet is sent to the sink node. The sink node checks if the *Energy_Consumption* is smaller than M_{PCost} , if so, the M_{PCost} and the corresponding *ID_List* will be updated. When *round* is multiple of *roundToUpdate*, the sink node will perform global deposition process. Each node in the *ID_List* will add additional M_{PCost} pheromones as in equation (12). Therefore, ES-DAACA can increase the amount of pheromones of those nodes who locate in

the path with the minimum energy consumption. More optimal solutions of data aggregation paths will be likely to be built. Obviously ES-DAACA will enhance the performance of Basic-DAACA.

$$\eta(i, j) = \eta(i, j) + M_{PCost}(j \in Nb(i), i \in P_{min}, j \in P_{min}) \quad (12)$$

Therefore, ES-DAACA will enhance the probability of finding global optimization solution, which preserves the energy and prolongs the network lifetime.

5.4. Maximum & Minimum based DAACA (MM-DAACA)

To limit the bound of the pheromones, we impose two constants to represent the maximum and minimum bounds of pheromones. We name this heuristic algorithm as Maximum & Minimum based DAACA (MM-DAACA for short).

In MM-DAACA, all the steps are identical with ES-DAACA, the only difference is that we use $[\eta_{Min}, \eta_{Max}]$ to limit the range of the pheromones. The reason is that if the pheromone is not limited in a range, some paths will own higher probabilities than the others, nevertheless, with the transmission going on, nodes in this kind of path will cost more energy than the others. But according to equation (3), the amount of pheromones is still large, which may cause them more likely to be selected as the next hop, and ultimately results in local optimal solution. However, if the constraints of pheromones are imposed, the aforementioned phenomenon can be avoided, variety of paths will be formed, hence, the global optimal solution will be more likely to be found.

5.5. Ant Colony System based DAACA (ACS-DAACA)

Similar to the ant colony system mentioned in section 3.2, we develop the Ant Colony System based DAACA (ACS-DAACA for short) algorithm for establishing the data aggregation paths to preserve energy.

In ACS-DAACA, when selecting the next hop node, the algorithm 5 is performed. q_0 is a parameter. After successfully sending the packet in each round,

Algorithm 5 Selecting the next hop of ACS-DAACA

- 1: Generate a random number q in $[0, 1]$.
 - 2: **if** $q \leq q_0$ **then**
 - 3: Select the neighbor node with the maximum $\tau(i, j)^\alpha \times \eta(i, j)^\beta$
 - 4: **else**
 - 5: select the next hop according to (1)
 - 6: **end if**
-

each node will locally update the pheromones as follows.

$$\eta(i, j) = (1 - \rho) \times \eta(i, j) + \rho \times M_{PCost}(j \in Nb(i), i \in P_{min}, j \in P_{min}) \quad (13)$$

Since the result of equation (13) is proved to be constrained within a certain range [16], therefore, it is similar to MM-DAACA which can maintain the variety

of the solutions. What's more, in ACS-DAACA, we propose local pheromone updating strategy as :

$$\eta(i, j) = (1 - \zeta) \times \eta(i, j) + \zeta \times \Delta\eta(i, j)^{best} \quad (14)$$

ζ stands for the rate of evaporating pheromones. Where

$$\Delta\eta(i, j)^{best} = \min_{j \in Nbr(i)} (\eta(i, j)) \quad (15)$$

The local pheromones updating aims at reducing the probability of the selected node, thus the probabilities of unselected nodes will increase. Consequently, ACS-DAACA will generate more aggregation paths than MM-DAACA through which global optimal solution will be more likely to be found.

6. Simulation Results and Analysis

In this section, several simulations are performed to evaluate and compare the performances of the DAACA family with other data aggregation algorithms. We use Java to develop a simulation platform to analyze the energy cost of each node and the network lifetime. We have testified the accuracy of the platform by simulating the third set of experiments in [14] and get identical statistics on the platform. Then we simulate different algorithms(i.e. LMST, PEDAP, PEDAP-PA, L-PEDAP, ACA, Basic-DAACA, ES-DAACA, MM-DAACA, ACS-DAACA) on the platform. In the experiments, multiple sources periodically send data to the sink node. Related parameters are shown in table 3 which are identical with those in [14]. We have run millions of experiments to highlight the advantages of DAACA in this paper. What's more, features of network topology are discussed. Furthermore, the time complexity analysis and characteristic analysis are given.

6.1. Average Energy Cost

All the nodes are deployed randomly as (Length \times Width, Node Number). Different numbers of packets are transmitted from the source nodes to the sink node. We calculate the average remaining energy of nodes with different algorithms. Higher remaining energy means less energy cost. Figure 2 - figure 4 illustrate the comparison results. Firstly each algorithm will initialize the topology of the network. With the transmissions going on, some of the algorithms will re-construct (or maintain) the topology to balance the energy cost of each node [17]. Hence, the topologies of some algorithms are dynamical.

The mechanism of establishing network topology of LMST and L-PEDAP are nearly identical, the energy consumptions of construction and maintenance of the network topology are high. Because each node in the network establishes the minimum spanning tree in the neighborhood locally. Then local minimum spanning trees connect into the topology of the network. When establishing the local minimum spanning tree, the weight of edge is directly related to the remaining energy of the node. Hence, each node needs to communicate with

Table 3: Experimental Parameters

| Item | Parameters |
|--|---|
| Algorithm | LMST, PEDAP, PEDAP-PA, L-PEDAP, ACA, Basic-DAACA, ES-DAACA, MM-DAACA, ACS-DAACA |
| Packet Size (Length×Width, Node Number) | 4098 bits (40 × 50, 200), (60 × 60, 400), (70 × 80, 600), (80 × 90, 800), (100 × 100, 1000), (100 × 120, 1200), (100 × 140, 1400), (120 × 130, 1600), (130 × 120, 1800), (140 × 140, 2000) |
| Node Distribution | Random |
| Initial Node Energy | 10J |
| Packets Number | 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 100000 |
| Number of Sources | 10 |
| R | 10 |
| α, β | $\alpha = 2, \beta = 2$ |
| Initial Pheromone | 0.8 |
| η_{Min}, η_{Max} | $\eta_{Min} = 0.5, \eta_{Max} = 0.9$ |
| ρ | $\rho = 0.2$ |
| $roundToUpdate$ | 100, 200 |
| ζ | 0.9 |
| q_0 | 0.5 |
| ρ in ACA | 0.3 |

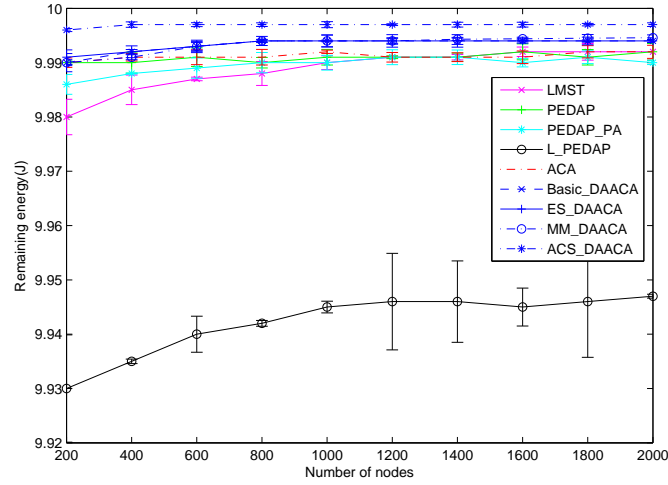


Figure 2: Average Energy Cost after Transmitting 10000 Packets

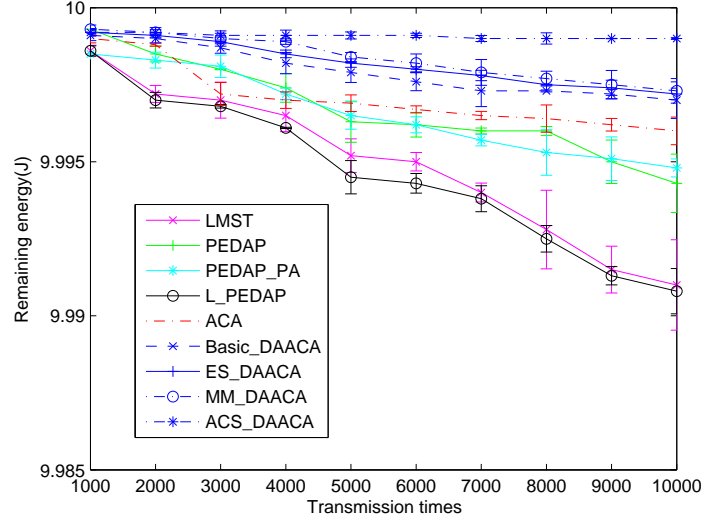


Figure 3: Average Energy Cost of the Network with the size $(140 \times 140, 2000)$

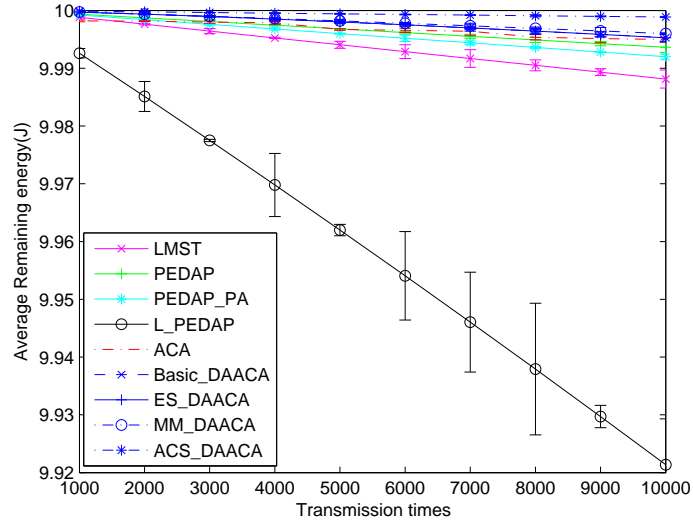


Figure 4: Average Results of all the Experiments of Average Energy Cost

neighbors to obtain their remaining energies periodically to update the weights of edges locally, frequent communications cast a heavy burden of all the nodes in the network. Therefore, the LMST and L-PEDAP cost more energy than others.

In ACA, the energy cost concentrates on the frequent depositions of pheromones. Because whenever a packet is received by a node, if the TTL of the packet is bigger than the EHC (Extended Hop Count)[14] of that node, the deposition of pheromones will start. The feedback message will be transmitted to inform the upstream nodes to deposit pheromones. Suppose there are a large number of packets transmitting in the network, and $TTL \gg EHC$, the feedback messages will be massively used, which consumes considerable energy. Therefore the energy consumption mainly concentrates on too frequent transmitting feedback messages.

In PEDAP and PEDAP-PA, the Prim algorithm are used to construct the topology and maintain the topology from the global view. Such methods can save the energy of communicating with neighbor nodes, but the sink node will suffer from more energy consumptions than the other nodes, because this is a centralized version of method. The sink node is responsible for establishing the topology of the network and informing all the nodes about the topology of the network via broadcasting. The energy cost of broadcast is high, especially when some nodes locate far away from the sink node. Whenever the network topology is changed, the broadcasting will be executed. Therefore, the energy costs of PEDAP and PEDAP-PA in constructing and maintaining the network topology are high.

In the family of DAACA, there is no energy consumption in maintaining the topology or reconstructing the MST, therefore, the average energy cost is lower than other algorithms. The energy consumption concentrates on sending packets to the next hop. After certain rounds of transmissions, the global pheromones adjustment will be carried out, which does not cost much energy. From the figure 2 - figure 4 we can see, the relationship among the four methods of the DAACA family in terms of average energy cost is $ACS < MM < ES < Basic$. MM-DAACA, ES-DAACA and ACS-DAACA enhance the quality of the solutions of Basic-DAACA which generate more energy preservation data aggregation paths for the network and eventually save the energy.

6.2. Energy Difference

In this section, we calculate energy difference to evaluate the balanced consumption of energy. We aims at testifying which method can balance the usage of energy. Because the balanced consumption can directly affect the lifetime of the network, which is the most prominent feature in energy efficiency in wireless sensor networks. If the energy difference is big, the energy is not averagely used whereas smaller energy difference indicates the balanced energy consumption.

In figure 5 - figure 7, we can see that the energy difference of L-PEDAP is the highest, the reason is that in L-PEDAP, each node needs to establish local minimum spanning tree periodically. L-PEDAP is energy aware, the weights of edges in the neighborhood always change, which demands each node to set up

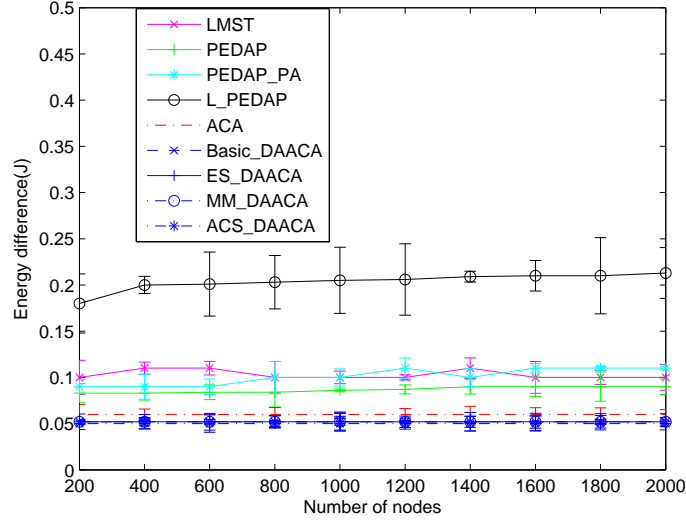


Figure 5: Energy Difference after Transmitting 10000 Packets

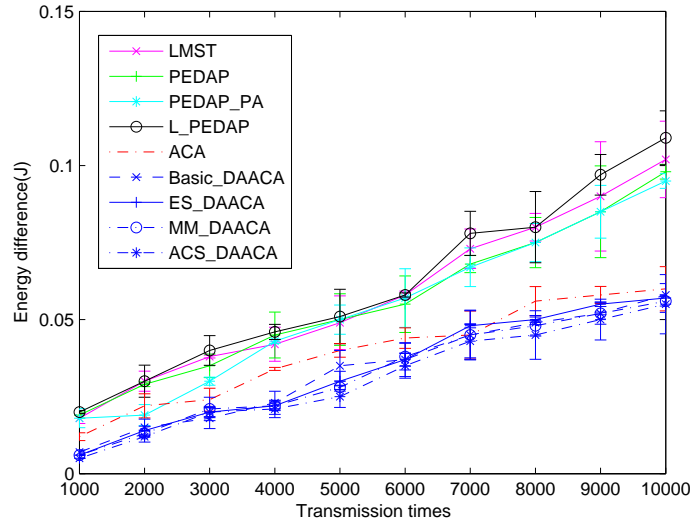


Figure 6: Energy Difference of the Network with the size (140 x 140, 2000)

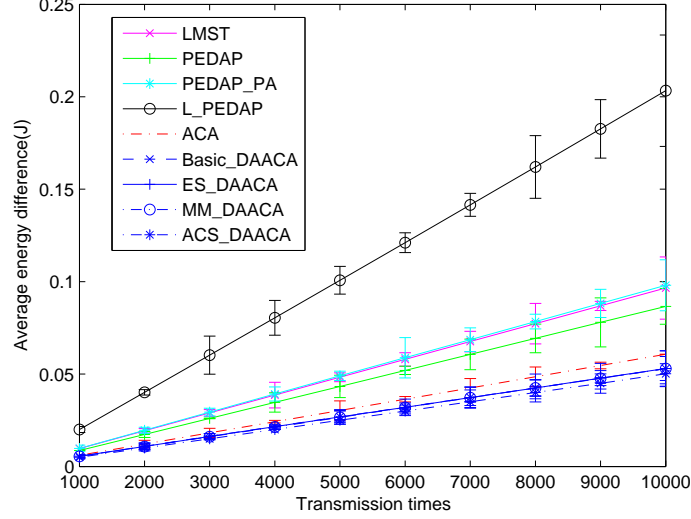


Figure 7: Average Results of all the Experiments of Average Energy Cost Energy Difference

the local minimum spanning tree repeatedly. Although L-PEDAP can balance the consumption of energy within neighborhood, a node with bigger neighbor set will actually consumes more energy than the smaller one. Because in the dense region of the network, each node owns many neighbor nodes. When establishing the LMST, each node needs to communicate with all the neighbors to acquire the energy information of the neighbors. Therefore, a node with large neighbor node set will spend much energy in establishing LMST, whereas a node owns small neighbor set cannot spend much energy, which yields unbalanced energy consumption such two kinds of nodes. Therefore, a big energy difference occurs.

Although LMST utilize RNG to establish the network topology, the same problem of L-PEDAP still exists, which leads to a big energy difference.

In PEDAP and PEDAP-PA, the sink node consumes most of the energy, because it needs to broadcast the topology of the network to all nodes in the network periodically. Thereby, there is a large energy difference between the sink node and others.

Whereas in DAACA, it takes the advantages of both centralized and decentralized algorithms. When selecting the next hop, each node will estimate the remaining energy of its neighbor to achieve balanced usage of energy. Moreover, in the process of depositing global pheromones, the path with minimum energy cost will obtain additional pheromones, by which the balanced usage of energy is implemented globally. The results show that DAACA can balance the energy cost of the network. The energy difference relationship among DAACA family is Basic>MM>ES>ACS.

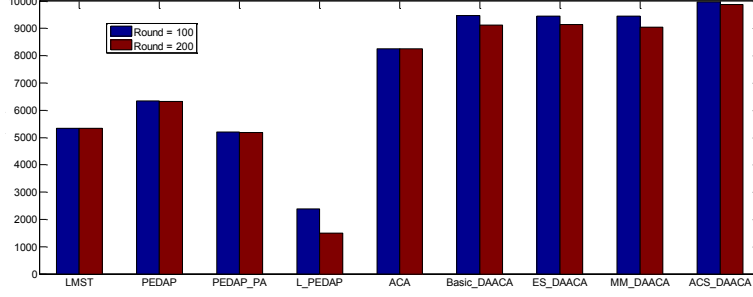


Figure 8: Lifetime of Network

6.3. Network Lifetime

The initial energy of each node is set as 0.05J. We measure how many rounds the network will sustain until any node exhausts in each algorithm. The parameter *roundToUpdate* is set as 100 and 200 respectively. From figure 8, the DAACA family shows longer lifetime than the other algorithms. And the relationship within the DAACA family is $ACS > MM > ES > Basic$. In PEDAP and PEDAP-PA, the sink node will firstly exhausted because of frequent broadcasting of topology. The lifetime of L-PEDAP and LMST are short which mainly results from massive energy costs in establishing local minimum spanning tree. In ACA, the energy cost mainly concentrates on some shortest paths. DAACA can guarantee the balanced energy costs of nodes by adjusting quantities of pheromones, therefore, it can prolong the lifetime of the network.

6.4. Successful Ratio of One Hop Transmission

In this section, we come to analyze the average success ratio of one hop transmission of each algorithm. We choose the data set $(100 \times 100, 1000)$. Packets are sent from the sources to the destination nodes. We count the success ratio of one hop transmission of each packet and calculate the average successful ratio of the one hop transmission. In figure 9, the successful ratio of DAACA is higher than other algorithms. The relationship within the DAACA family is $ACS > MM > ES > Basic$.

6.5. Network Topology Analysis

In this section, we analyze the performance of network topology in terms of node average degree and average transmission radius. As the experiment in [12], the data set $(100 \times 100, 1000)$ is chosen to run the simulation.

6.5.1. Average Degree

In this experiment, we come to analyze the average degree of each node. The average degree is calculated as the number of edges divided by the number of nodes. Bigger average degree indicates the higher spatial reuse[12].

Table 4 lists the average degree of each node in the network. In PEDAP and PEDAP-PA, the Prim algorithm is used to generate the minimum spanning

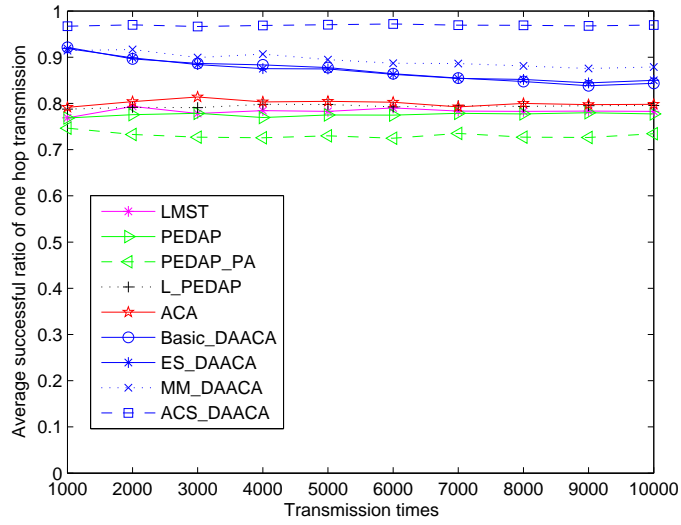


Figure 9: Average Successful Ratio of One Hop Transmission

Table 4: Average Degree

| Algorithm | Average Degree |
|------------------|----------------|
| LMST | 2.04 |
| PEDAP & PEDAP-PA | 0.999 |
| L-PEDAP | 2.04 |
| ACA | 0.999 |
| DAACA | 0.999 |

tree of the network globally, since in the minimum spanning tree, the number of edge equals to the number of nodes minus 1, therefore the sum of the degree equals to the number of nodes minus 1 and the average degree is $n - 1/n$ where n is the number of nodes in the network. In ACA and DAACA, although the topologies are dynamic, each time only one node is selected as the next hop until the packet arrives at the sink node. Therefore the total degree still equals to that of PEDAP and PEDAP-PA. Whereas in LMST and L-PEDAP, local MST is produced, which cause many loops in the topology, in such circumstance the number of edges is bigger than n . Hence, the average degree is higher than other algorithms.

In a word DAACA can achieve a higher spatial reuse.

6.5.2. Average Transmission Radius

In this experiment, we calculate the average transmission radius of each node. As illustrated in [12], average transmission radius is another factor which indicates the spatial reuse.

Table 5: Average Transmission Radius

| Algorithm | Radius |
|------------------|--------|
| LMST | 4.81 |
| PEDAP & PEDAP-PA | 2.19 |
| L-PEDAP | 2.07 |
| ACA | 1.8 |
| DAACA | 1.79 |

As illustrated in [12], the smaller average transmission radius will lead to a higher spatial reuse. In table 5, the results of the average transmission radius of each algorithm are listed. In LMST and L-PEDAP, the local MST are established, more links will be used to establish the topology, which is the main reason of causing the higher transmission radius. In PEDAP and L-PEDAP, the average degree is 2.19 and 2.07 respectively. In ACA and DAACA, the next hop node is selected which is nearer to the sink node than to the current node. The selection is based on the probabilities; the shorter edge has a higher possibility to be chosen. In DAACA the edge is selected based on distance and energy, therefore, the radius is a little shorter than ACA.

From the topology analysis, we can conclude that DAACA achieves a higher spatial reuse than the other algorithms.

6.6. Time Complexity Analysis

In this section, we mainly analyze the complexity of constructing and maintaining the topology. Because in the initialization phase, the steps (e.g. initialize the neighbor set, initialize routing protocol, et al.) are nearly identical, we only compare the time complexity in constructing and maintaining the topology of each method.

6.6.1. Time Complexity of Topology Construction

Table 6: Complexity of Topology Construction

| Algorithm | Time Complexity |
|---------------------|-----------------|
| LMST | $O(n \log(n))$ |
| PEDAP-PA & LEDAP-PA | $O(e \log(n))$ |
| L-PEDAP | $O(n \log(n))$ |
| ACA | $O(n^2)$ |
| DAACA | $O(n^2)$ |

Table 6 reveals the time complexity of each algorithm in constructing network topology. In LMST, each node establishes the local minimum spanning tree, the Prim algorithm is utilized to generate the MST, the complexity to set up MST is $O(e \log(n))$, where e is the number of links within the visible neighborhood [13] and $e \geq n$. But LMST is only apt to the network with a small number of nodes or low density network. Because when LMST is taken upon numerous nodes, each node will own a large set of neighbors which yields to $e \approx n^2$. Therefore the overall complexity in constructing LMST is high. Whereas in PEDAP and PEDAP-PA, the centralized version of minimum spanning tree is established, which owns a complexity of $O(e \log(n))$. Compared with LMST, the complexity is relatively low, but it is a centralized method which casts a heavy burden on the sink node. In L-PEDAP, the topology construction is identical with LMST; therefore, the complexity is still high. In ACA, when establishing the network topology, each node must broadcast messages to his neighbors in order to inform them its hops to the sink node. Therefore, the complexity is $O(n^2)$. Topology construction DAACA is nearly the same as ACA, thus the complexity is $O(n^2)$.

6.6.2. Time Complexity of Topology Maintenance

In this section, the complexity of maintaining the topology is discussed. Table 7 lists the complexity of each algorithm. Since each method updates its topology in every *roundToUpdate* rounds periodically, it is feasible to compare the complexity in just one period. In LMST, PEDAP, PEDAP-PA and L-PEDAP, all the topologies are required to be re-constructed. Thus the complexities are the identical with those in constructing topologies of section 6.6.1. In ACA, whenever a node meets the condition of $EHC \leq TTL$, operations of pheromones are performed, some extra control messages [14] are needed to be transferred, therefore, the complexity of ACA is $O(rn)$ (r stands for *roundToUpdate*). Whereas in the family of DAACA, each node locally calculates the probability of selecting the next hop in every round. Only one broadcasting message for updating the energy evaluation is required, therefore, the complexity is $O(n)$. From table 7, it can be concluded that the complexity of maintaining the topology of DAACA is lower than the other methods.

6.7. Other Characteristic

In this section, we analyze other characteristics of DAACA.

6.7.1. Robustness

When selecting the next hop, each node will refer to the probability recorded in the routing table, all the path are dynamical, if a node is removed from the network, only the nodes nearby are required to update the routing table. Therefore, the adjustment of removing a node can be achieved locally, therefore, DAACA owns the feature of robustness.

6.7.2. Fault Tolerance

Suppose errors occur in the element of the node which prohibits it from working. In this case, the node nearby can remove this node from the routing table. Therefore, the faulty nodes will no longer exist in the topology of the network. Hence they cannot affect the transmission of the network. And we can conclude that DAACA is fault tolerant to faulty nodes.

6.7.3. Scalability

Since nodes are deployed randomly in the network, and each node only needs to maintain the routing table for dynamically selecting the next hop. Therefore, if a node is added in the network, it only needs to broadcast his identity information and sets up routing table, the nodes nearby are only required to add a new item of the routing table. So we can conclude that DAACA is scalable.

7. Conclusion

In this paper, a family of ant colony algorithms called DAACA for data aggregation has been presented which contains three phases: the initialization, packet transmission and operations on pheromones. After initialization, each node estimates the remaining energy and the amount of pheromones to compute the probabilities used for dynamically selecting the next hop. After certain rounds of transmissions, the pheromones adjustment is performed periodically, which combines the advantages of both global and local pheromones adjustment for evaporating or depositing pheromones. Four different pheromones adjustment strategies are designed to achieve the global optimal network lifetime,

Table 7: Complexity of Topology Maintenance

| Algorithm | Time Complexity |
|---------------------|-----------------|
| LMST | $O(n \log(n))$ |
| PEDAP-PA & LEDAP-PA | $O(e \log(n))$ |
| L-PEDAP | $O(n \log(n))$ |
| ACA | $O(rn)$ |
| DAACA | $O(n)$ |

namely Basic-DAACA, ES-DAACA, MM-DAACA and ACS-DAACA. Compared with some other data aggregation algorithms, DAACA shows higher superiority on average degree of nodes, energy efficiency, prolonging the network lifetime, computation complexity and success ratio of one hop transmission. At last we analyze the characteristic of DAACA in the aspects of robustness, fault tolerance and scalability.

ACKNOWLEDGEMENTS

This work was partially supported by the National Natural Science Foundation of China under Grants No.60703101 and No.60903153, and the Fundamental Research Funds for the Central Universities.

References

- [1] G. Pottie, Wireless sensor networks, in: Information Theory Workshop, 1998, IEEE, 2002, pp. 139–140.
- [2] F. Lewis, Wireless sensor networks, Smart environments: technologies, protocols, and applications (2004) 1–18.
- [3] S. Roundy, D. Steingart, L. Frechette, P. Wright, J. Rabaey, Power sources for wireless sensor networks, *Wirel. Sens. Netw.* (2004) 1–17.
- [4] S. Muruganathan, D. Ma, R. Bhasin, A. Fapojuwo, A centralized energy-efficient routing protocol for wireless sensor networks, *Commun. Mag. IEEE* 43 (3) (2005) 8–13.
- [5] O. Younis, S. Fahmy, HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks, *IEEE T. Mob. Comp.* 3 (4) (2004) 366–379.
- [6] O. Younis, S. Fahmy, Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach, in: INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 1, IEEE, 2004, pp. 1–12.
- [7] S. Bandyopadhyay, E. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, Vol. 3, IEEE, 2003, pp. 1713–1723.
- [8] H. Chen, C. Tse, J. Feng, Minimizing effective energy consumption in multi-cluster sensor networks for source extraction, *IEEE T. Wirel. Comm.* 8 (3) (2009) 1480–1489.
- [9] K.-W. Fan, S. Liu, P. Sinha, Data aggregation in wireless sensor networks, in: *Wireless Sensor Networks and Applications*, Springer US, 2008, pp. 331–347.

- [10] C. Savarese, J. Rabaey, K. Langendoen, Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks, in: Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference, USENIX Association, 2002, pp. 317–327.
- [11] H. Tan, I. Korpeoglu, Power efficient data gathering and aggregation in wireless sensor networks, *ACM SIGMOD Record* 32 (4) (2003) 66–71.
- [12] N. Li, J. Hou, L. Sha, Design and analysis of an MST-based topology control algorithm, *IEEE T. Wirel. Comm.* 4 (3) (2005) 1195–1206.
- [13] H. Tan, I. Korpeoglu, I. Stojmenovic, Computing Localized Power Efficient Data Aggregation Trees for Sensor Networks, *IEEE Tran. Paral. Dist. Syst.* 1 (99) (2010) 1–1.
- [14] W. Liao, Y. Kao, C. Fan, Data aggregation in wireless sensor networks using ant colony algorithm, *J. Netw. Comp. Appl.* 31 (4) (2008) 387–401.
- [15] J. Jaromczyk, G. Toussaint, Relative neighborhood graphs and their relatives, *Proceedings of the IEEE* 80 (9) (2002) 1502–1517.
- [16] M. Sharaf, J. Beaver, A. Labrinidis, P. Chrysanthis, Balancing energy efficiency and quality of aggregate data in sensor networks, *VLDB J.* 13 (4) (2004) 384–403.
- [17] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proceedings of the 7th annual international conference on Mobile computing and networking, ACM, 2001, pp. 70–84.
- [18] M. Dorigo, T. Stutzle, *Ant colony optimization*, MIT press, 2004.
- [19] J. Kennedy, *Swarm intelligence, Handbook of Nature-Inspired and Innovative Computing* (2006) 187–219.
- [20] M. Dorigo, L. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *Evolutionary Computation, IEEE Transactions on* 1 (1) (1997) 53–66.
- [21] P. Toth, D. Vigo, *The vehicle routing problem*, Vol. 9, Society for Industrial Mathematics, 2002.
- [22] S. Okdem, D. Karaboga, Routing in Wireless Sensor Networks Using an Ant Colony Optimization (ACO) Router Chip, *Sensors* 9 (2) (2009) 909–921.
- [23] T. Camilo, C. Carreto, J. Silva, F. Boavida, An energy-efficient ant-based routing algorithm for wireless sensor networks, *Ant Colony Optimization and Swarm Intelligence* (2006) 49–59.

- [24] R. Misra, C. Mandal, Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks, in: *Wireless and Optical Communications Networks*, 2006 IFIP International Conference on, IEEE, 2006, pp. 5–pp.
- [25] C. Intanagonwiwat, D. Estrin, R. Govindan, J. Heidemann, Impact of network density on data aggregation in wireless sensor networks, in: *Distributed Computing Systems*, 2002. Proceedings. 22nd International Conference on, IEEE, 2002, pp. 457–458.
- [26] C. Intanagonwiwat, D. Estrin, R. Govindan, J. Heidemann, Impact of network density on data aggregation in wireless sensor networks, in: *Distributed Computing Systems*, 2002. Proceedings. 22nd International Conference on, IEEE, 2002, pp. 457–458.
- [27] K. Kalpakis, K. Dasgupta, P. Namjoshi, Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks* 1, *Comp. Netw.* 42 (6) (2003) 697–716.
- [28] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, *Comp. Netw.* 38 (4) (2002) 393–422.
- [29] F. Neumann, C. Witt, Runtime analysis of a simple ant colony optimization algorithm, *Algorithmica* 54 (2) (2009) 243–255.
- [30] S. Madden, R. Szewczyk, M. Franklin, D. Culler, Supporting aggregate queries over ad-hoc wireless sensor networks, in: *Mobile Computing Systems and Applications*, 2002. Proceedings Fourth IEEE Workshop on, IEEE, 2002, pp. 49–58.
- [31] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B. Krogh, Energy-efficient surveillance system using wireless sensor networks, in: *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, ACM, 2004, pp. 270–283.
- [32] G. Mao, B. Fidan, B. Anderson, Wireless sensor network localization techniques, *Comp. Netw.* 51 (10) (2007) 2529–2553.
- [33] A. Coloni, M. Dorigo, V. Maniezzo, et al., Distributed optimization by ant colonies, in: *Proceedings of the first European conference on artificial life*, Vol. 142, 1991, pp. 134–142.
- [34] T. Camilo, C. Carreto, J. Silva, F. Boavida, An energy-efficient ant-based routing algorithm for wireless sensor networks, *Ant Colony Optimization and Swarm Intelligence* (2006) 49–59.
- [35] N. Ding, X. Liu, et al., Data gathering communication in wireless sensor networks using ant colony optimization, in: *Robotics and Biomimetics*, 2004. ROBIO 2004. IEEE International Conference on, IEEE, 2005, pp. 822–827.

